

Troubleshooting

From time to time, Bcfg2 produces results that the user finds surprising. This can happen either due to bugs or user error. This page describes several techniques to gain visibility into the bcfg2 client and server and understand what is going on.

1. Figure out if the malfunction is on the client or server side.
Cache a copy of the client using `bcfg2 -c /tmp/config.xml`. This file contains all aspects of client configuration. This file is structured as a series of bundles and base entries. Search for the entry of interest. If it is correct, then the client is at issue.
2. Review server log messages.
The `bcfg2-server` process logs to syslog facility `LOG_DAEMON`. The server produces a series of messages upon a variety of events and errors.
3. Check if all repository XML files conform to schemas.
Bcfg2 comes with XML schemas describing all of the XML formats used in the server repository. A validation command, `bcfg2-repo-validate`, is included with the source distribution and all packages. Run it with the `-v` flag to see each file and the results of its validation.
4. If the `bcfg2` server is not reflecting recent changes, try restarting the `bcfg2-server` process.
If this fixes the problem, it is either a bug in the underlying file monitoring system (fam or gamin) or a bug in Bcfg2's file monitoring code. In either case, file a [ticket](#) in the tracking system. In the ticket, include:
 - ◆ Filesystem monitoring system (fam or gamin)
 - ◆ kernel version (if on linux)
 - ◆ if any messages of the form "Processed N <fam/gamin> events in M seconds" appeared between the modification event and the client configuration generation request appeared in the server log
 - ◆ Which plugin handled the file in the repository (Cfg, Svcmgr, Rules, Pkgmgr, TCheetah, Metadata)
 - ◆ If a touch of the file after the modification causes the problem to go away
5. Bcfg2 server operations can be simulated using the `bcfg2-info` command.
The command is interactive, and has commands to allow several useful operations
 - ◆ Current client metadata (profile and group) settings -- clients
 - ◆ Current group metadata values -- groups
 - ◆ Configuration entries provided by plugins -- mappings
 - ◆ Build a config file for a client -- `buildfile <filename> <hostname>`
 - ◆ Build the abstract configuration (list of entries) for a client -- `showentries <client> <type>`
 - ◆ Build the complete configuration for a client -- `build <hostname> <output-file>`